

Dr. Gerhard Friedrich, Mag. Christian Knogler

Die Pflichten des Anwenders bei Prototyping und Rapid Application Development¹

1.	Worum geht es?	2
2.	Der Erfolg des EDV-Einsatzes wird im Umfeld der Software-Entwicklung entschieden	3
2.1.	Strategie und Informatik.....	3
2.2.	Organisation und Informatik.....	4
2.3.	Motivation und Informatik.....	7
2.4.	Projektmanagement und Informatik.....	7
3.	Was mit Prototyping und RAD einfacher wird?.....	8
4.	Was trotz Prototyping und RAD schwierig bleibt?	8
5.	Gute Architektur kommt nicht von Gelegenheitsarchitekten	8

¹ Beitrag zum ADV-Tagung CASE 93, die sich grundlegenden methodischen Fragen des Computer Aided Software Engineering widmete.

1. Worum geht es?

Neue Methoden und Werkzeuge verändern das Interaktionssystem von Anwendern und Entwicklern. Neue technische Möglichkeiten, veränderte Aufwandsrelationen für bestimmte Aufgaben eines Softwareentwicklungsprojektes schaffen neue Möglichkeiten oder gar Notwendigkeiten, ein Software-Projekt abzuwickeln.

Solange es aufwendig war (und vielfach noch ist), ein Bildschirm- oder Listen-Layout zu verändern, mussten diese Vorgaben möglichst früh im Projekt festgeschrieben werden. Ist es hingegen auch in späten Projektphasen leicht möglich, sogar neue Felder hinzuzufügen (die in der Datenbank schon vorhanden sind, bisher aber in der betreffenden Transaktion nicht verwendet wurden oder überhaupt neue Attribute), muss man sich mit diesen Vorgaben weniger Mühe geben. Der Anwender muss nicht mehr Layouts auf Papier erstellen (lassen) und dann unterschreiben, damit er nur mehr bei Strafe beträchtlicher Mehrkosten, die zu seinen Lasten gehen, Änderungswünsche geltend machen kann.

Geht es aber in diesen Punkten sozusagen etwas lockerer zu, wo noch kann man auf harte Arbeit und Verbindlichkeit in frühen Projektphasen verzichten? Braucht man Entity-Relationship-Diagramme, wenigstens einen Datenkatalog und wie lange kann man dort die Angaben über Typ und Feldlänge offen halten? Ab wann muss (wenn überhaupt) ein Funktionenkatalog festgelegt werden, wie fein muss er gegliedert und wie detailliert muss er beschrieben sein?

Kann es sich vielleicht der Anwender angesichts der neuen Entwicklungswerkzeuge und -plattformen überhaupt bequem machen und sich mit Informationen über seine Anforderungen in freier Prosa sowie kritischen Anmerkungen zu den ihm vorgestellten Prototypen begnügen und darauf warten, dal im die Softwareentwickler früher oder später das erträumte Informationssystem liefern?

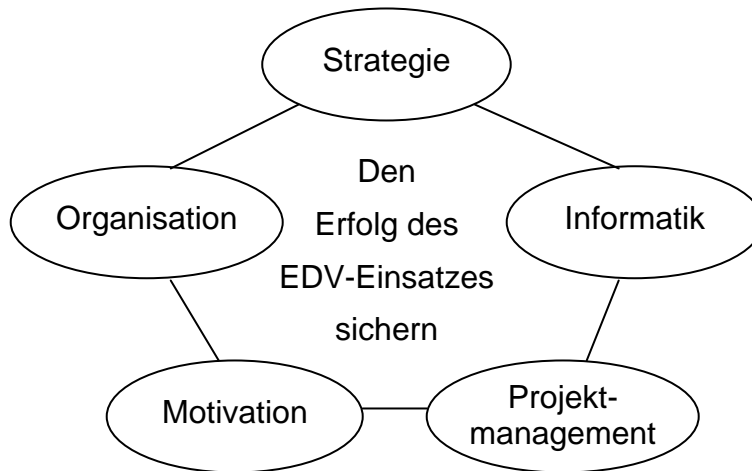
Oder gilt umgekehrt für die Softwareentwickler, dass sie nur noch dem Anwender ein Werkzeug in die Hand geben und es ihm erklären müssen, ihn damit einige Zeit arbeiten lassen und zum Schluss die abgelieferten Teile zu einem benutzerfreundlichen, fachlich optimalen und kostengünstigen Informationssystem zusammenfügen?

Natürlich wissen wir alle, dass beide Extreme nicht realisierbar sind und natürlich gibt es niemanden, der ernsthaft ein solches Szenario vertritt. Aber doch ist einiges in Fluss geraten und damit verbunden gibt es eine Verunsicherung bei allen Beteiligten.

Wir konzentrieren uns in unserem Beitrag auf die Rolle des Anwenders in modernen Entwicklungsumgebungen, die mit den Leitbegriffen dieser Tagung, nämlich Prototyping und Rapid Application Development (RAD) beschrieben sind. Wie schon der Titel unseres Referates signalisiert, meinen wir, dass der Anwender auch bei Nutzung der modernen Entwicklungsplattformen weiterhin viel zu tun hat und diese Aufgaben sogar unbedingt erfüllen muss, wenn Softwareentwicklung den gewünschten Nutzen bringen soll.

2. Der Erfolg des EDV-Einsatzes wird im Umfeld der Software-Entwicklung entschieden

Ein Informationssystem ist in die Strukturen eines Unternehmens eingebettet. Ob es die Erwartungen erfüllt, hängt davon ab, wie gut es in das Gesamtsystem passt. Wir sprechen von einem Pentagramm, das den Erfolg des EDV-Einsatzes bestimmt.

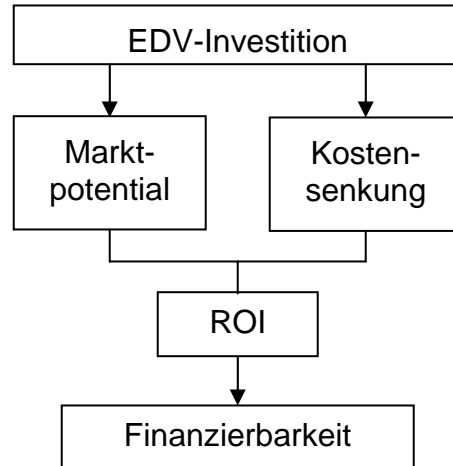


Zwischen allen diesen Elementen bestehen Wechselwirkungen und die Veränderung der Methoden und Werkzeuge der Software-Erstellung, die auf dieser Tagung im Mittelpunkt steht, wirkt sich daher ebenfalls auf alle anderen erfolgskritischen Elemente aus.

2.1. Strategie und Informatik

Die **Unternehmensstrategie** (ob sie nun explizit formuliert und bewusst umgesetzt wird oder sich nur aus der Gesamtheit der Handlungen erschließen lässt) gibt an, in welchen Märkten das Unternehmen mit welchen Produkten und Dienstleistungsangeboten erfolgreich sein will. Mit der Strategie eng verbunden sind Annahmen über die strategischen Erfolgsfaktoren, also jene Gegebenheiten in und außerhalb des Unternehmens, die über Erfolg und Misserfolg entscheiden. Weiters beinhaltet ein strategisches Programm auch Annahmen über Art und Größe des Wachstums verschiedener Unternehmensbereiche.

Der Anwender muss eine klare Vorstellung davon entwickeln, welche Beiträge EDV-Systeme zum Erfolg des Unternehmens leisten sollen. Der klassische Bereich der Kostensenkung ist nach wie vor wichtig, aber immer mehr tritt der Beitrag des EDV-Einsatzes zur Verbesserung des Kundenservice und zu einer erhöhten Verfügbarkeit von Kunden- und Marktdaten in den Vordergrund der Überlegungen. Aus beiden Komponenten in Kombination ergibt sich die Rentabilität einer EDV-Investition und damit die finanzielle Machbarkeit und Zweckmäßigkeit.



Sicherlich kann und muss der EDV-Experte den Anwender auf neue technische Möglichkeiten hinweisen, so dass z.B. der Einsatz von grafischen System auf ultraleichten Notebooks mit Farbschirm und Multimedia-Fähigkeiten neue Möglichkeiten der Kundenberatung durch den Außendienst bietet (mit heutigem Stand: bieten wird). Letztlich bleibt es aber die alleinige Pflicht und Verantwortung des Anwenders, die Positionierung einer EDV-Anwendung im Rahmen der Unternehmensstrategie vorzunehmen und damit die Finanzierbarkeit einer EDV-Investition zu sichern.

Sogar wenn die Software-Entwickler die zugesicherte Funktionalität nicht zu den vereinbarten Terminen und Kosten liefern, wird der Anwender immer noch die Verantwortung für die falsche Auswahl seiner Lieferanten übernehmen müssen. Jede Methodik, Software kostengünstiger, schneller, risikoärmer und mit höherer Auftragstreue zu entwickeln, liegt daher im ureigensten Interesse der Anwender und damit zeigt sich wiederum ein weiterer Aspekt des Themas dieser Tagung.

Aber es gibt noch eine andere Wechselwirkung zwischen neuen Methoden der Softwareentwicklung und der Strategie. Das nunmehr mögliche iterative Vorgehen begünstigt die Entwicklung kreativer Lösungen, da strategische, organisatorische und technische Phantasie besser zusammenwirken können als im Rahmen eines klassischen, streng phasenorientierten Vorgehens nach dem Wasserfall-Modell.

2.2. Organisation und Informatik

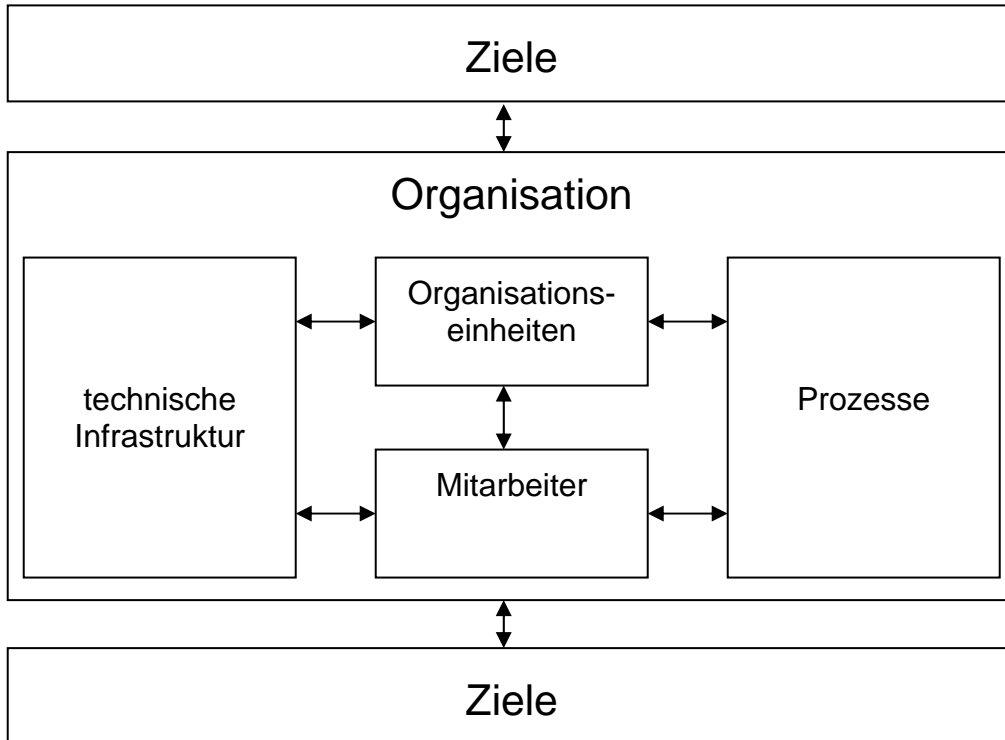
Es gehört schon zu den Stehsätzen der Software-Entwicklung, dass natürlich zuerst die organisatorischen Fragen geklärt werden müssen, bevor die EDV-Anwendung Sinn macht. Und es ist genauso wahr, dass in der Praxis regelmäßig versucht wird, über die EDV organisatorische Probleme in den Griff zu bekommen. Wenn eine Regel so oft ignoriert wird, besteht der Verdacht, dass sie entweder falsch oder zumindest in der Praxis nicht umsetzbar ist; gemäß einem pragmatischen Begriff von Richtigkeit ist sie damit aber auch im zweiten Fall falsch.

Wir orientieren uns bei unseren Überlegungen an dem auf der folgenden Seite grafisch dargestellten Modell von Organisationsplanung. Die Schnittstelle zur Informatik sind die Aufgaben, zu deren Erfüllung ein EDV-System einen Beitrag leisten soll.

Das nach wie vor unübertroffene Instrument zur Darstellung von organisatorischen Gegebenheiten ist unserer Erfahrung nach ein Funktionendiagramm, aus dem man klar erkennen kann, von welchen Stellen welche Aufgaben wahrgenommen werden. Alle wesentlichen Fragen der Modularisierung, der Möglichkeit von Stand-Alone-Lösungen bzw. die Notwendigkeit von Netzwerkanwendungen etc.

lassen sich auf dieser Grundlage beantworten.

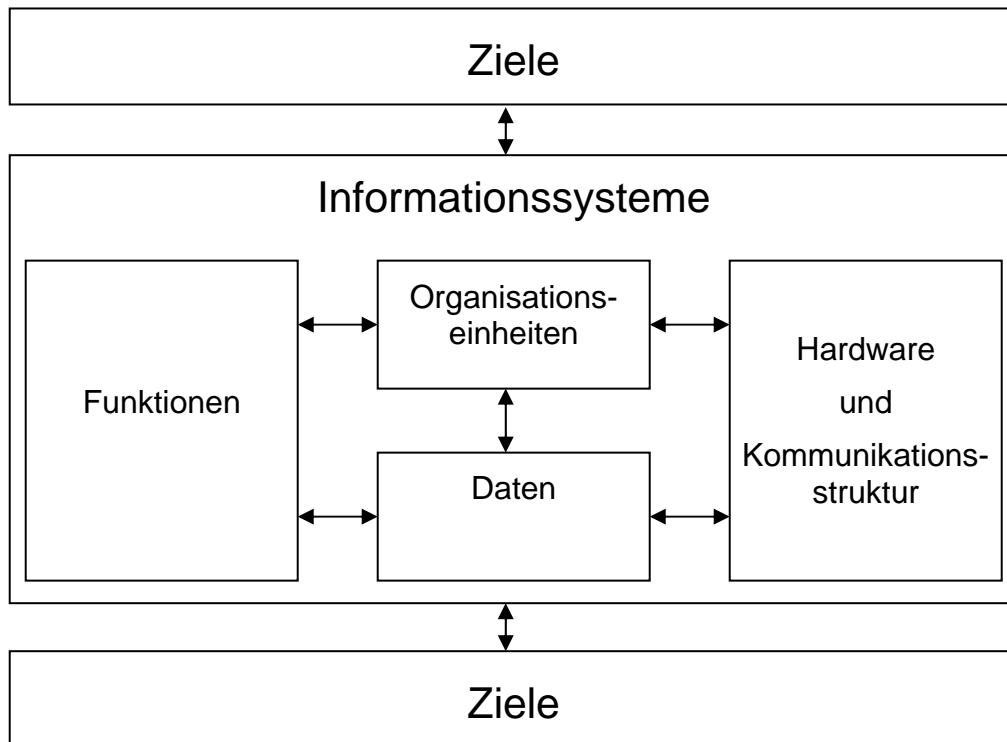
Organisationsplanung



Die grundlegende Weichenstellung für die Entwicklung einer EDV-Lösung ist die Auswahl jener Aufgaben, zu deren Wahrnehmung eine EDV-Lösung einen Beitrag leisten soll. Diese Beiträge werden in einem Funktionenmodell systematisch dargestellt.

Wir vertreten die Auffassung, dass für die Entwicklung operativer EDV-Systeme als Einstieg eine funktionale Sicht zweckmäßiger ist als eine datenorientierte Vorgangsweise. Dies gilt z.B. nicht für die Entwicklung von Management-Informationssystemen, aber die immer noch vielfach vertretene Auffassung, man müsste grundsätzlich mit der Entwicklung eines (möglichst unternehmensweit gültigen) Datenmodells in ein EDV-Projekt einsteigen, hat viele Projekte scheitern lassen. Wir wüssten übrigens auch nicht, wie man einen strikt datenorientierten Zugang mit Prototyping unterstützen könnte, hier geht es immer um ausgewählte Transaktionen, die modelliert und optimiert werden. Dass natürlich die Datenanalyse einen prominenten und erfolgskritischen Platz in jeder Software-Entwicklung einnimmt, soll hier nicht bestritten werden, es geht uns nur um die Prioritäten.

Analog der obigen Darstellung unseres Modells von Organisation sehen wir die Elemente einer Informationssystemplanung wie in nachfolgendem Modell dargestellt.

Informationssystemplanung

Organisation und Informationssysteme können natürlich separat optimiert werden, wobei man jeweils das andere Teilsystem konstant setzt. Dass dabei oft nur ein Suboptimum erreicht wird, steht aber ebenfalls außer Zweifel: Wüsste der Anwender, welche Möglichkeiten ihm die EDV bieten könnte, würde er sich vielleicht anders organisieren und wüsste der Softwareentwickler, welche Anforderungen der Anwender nicht genannt hat, weil er sie für nicht realisierbar hält, würde er manches anders lösen.

Wollte man grundsätzlich beide Teilsysteme gleichzeitig optimieren, würde man schnell die Grenzen der bewältigbaren Komplexität, der Beweglichkeit der Organisation und der Geduld der Anwender überschreiten. Genau hier scheint uns die Möglichkeit, die Phantasie des Anwenders in frühen Phasen mit Prototypen zu stimulieren ohne gleich alles in Frage zu stellen, von großem Nutzen.

Die letzte Entscheidung, bei welchen Aufgaben sich welche Art von EDV-Unterstützung mit welchem Grad an Komfort lohnt, kann aber wiederum nur der Anwender treffen und verantworten.

2.3. Motivation und Informatik

Das Problem der Akzeptanz von EDV-Lösungen existiert, seit nicht mehr Techniker für sich selbst programmieren, sondern es Anwender gibt, die mit den von Technikern entwickelten Softwaresystemen arbeiten (müssen). Wir verzichten aus guten Gründen darauf, hier einen Exkurs zum Thema Motivationspsychologie einzuschieben und begnügen uns mit einem sehr simplen Theorem:

Anwender sind motiviert, ein EDV-System zu nutzen, wenn sie sich davon mit genügend hoher Sicherheit eine Verbesserung ihrer Arbeitssituation erwarten.

Wenn jemand motiviert ist, die ihm von der Organisation zugedachten Aufgaben zu erfüllen, ist er zur Nutzung jedes EDV-Systems motiviert, das die Erfüllung dieser Aufgaben wirksam unterstützt. Fehlt die Motivation in bezug auf die Aufgaben, kann keine noch so "gute" EDV-Lösung mit Akzeptanz rechnen. Auch hier stoßen wir wieder auf einen unabdingbaren Beitrag des Anwenders (insbesondere der Führungskräfte der betroffenen Bereiche) zum Erfolg des EDV-Einsatzes

Prototyping hat auch hinsichtlich der Motivation der Anwender einen entscheidenden Vorteil. Durch die früh beginnende, regelmäßige Auseinandersetzung mit dem geplanten EDV-System werden einerseits die Erwartungen des Anwenders durch konkrete Informationen über die zu erwartende Funktionalität beeinflusst und da es ja das erklärte Ziel des Prototyping ist, die Anforderungen des Anwenders in den Entwurf einzuarbeiten, ist diese Beeinflussung der Erwartungen mit hoher Wahrscheinlichkeit positiv. Dass das berechtigte Gefühl, an der Entwicklung des EDV-Systems aktiv mitzuwirken, die Identifikation und damit Motivation noch zusätzlich fördert, sei noch ergänzend angemerkt.

Wir haben in unserer Projektarbeit beste Erfahrungen mit Workshops in frühen Projektphasen gemacht, zu denen wir ausgewählte Anwender eingeladen haben. Es ist natürlich wesentlich, dass nicht nur Fachleute, sondern auch Meinungsbildner und Multiplikatoren (die möglichst auch Experten sind), an diesen Workshops teilnehmen. Bei der Durchführung solcher Workshops ist natürlich ein Mindestmaß an Gesprächsführungs- und Moderationstechnik auf Seiten der EDV-Experten notwendig, nötigenfalls kann hier ein nicht allzu fachfremder Moderator beigezogen werden.

Ebenfalls für die Motivation der Anwender entscheidend ist die Gestaltung der Systemeinführung und der Schulung. Wir erarbeiten diese Pläne schon zu Beginn eines Projektes, denn aus den Möglichkeiten und Restriktionen der Systemeinführung (z.B. Aufrechterhaltung des laufenden Betriebes) und der Schulung (z.B. Verfügbarkeit von Trainern und Zeitbudget der zu schulenden Zielgruppe) ergeben sich oft genug entscheidende Anforderungen für die Gestaltung eines EDV-Systems.

2.4. Projektmanagement und Informatik

Gerade wenn man von einem streng phasenorientierten Vorgehensmodell abweicht (was de facto aber immer schon, wenn auch mit schlechtem Gewissen getan wurde), besteht die Gefahr, dass die Präzision der Projektplanung reduziert und das Projektmanagement lax wird. Es ist jedoch eine der Paradoxien, die wir aus den Projekten zur partizipativen Organisationsgestaltung gelernt haben, dass nicht nur mit zunehmender Zahl an Beteiligten sondern auch mit zunehmendem Grad an Partizipation das Projektmanagement umso straffer sein muss. Je mehr Gestaltungsspielraum man den Beteiligten einräumt, umso häufiger kommt es zu Veränderungen, die Auswirkungen in anderen Bereichen haben. Das Schnittstellenmanagement muss daher mit hoher Intensität wahrgenommen werden. Dabei müssen sowohl die Meilensteintermine aufeinander abgestimmt als auch der regelmäßige Informationsfluss gesichert werden.

Schriftliche und von allen Beteiligten verstandene und anerkannte Projektaufträge, eine klar definierte Projektorganisation, insbesondere ein Projektleiter mit ausreichenden Entscheidungsbefugnissen und Ressourcen sind einige weitere Elemente, die gutes Projektmanagement immer schon auszeichnen und unter den komplexeren Bedingungen des Prototyping umso wichtiger sind.

Es ist die nicht delegierbare Pflicht des Anwenders, insbesondere des Managements, die Gesamt-

verantwortung für den Erfolg und damit das Gesamt-Projektmanagement zu übernehmen.

3. Was mit Prototyping und RAD einfacher wird?

Es gibt kein Pflichtenheft, dass in allen Punkten "halt" und gerade die Gestaltung der Anwenderschnittstelle lebt von den Erfahrungen der Arbeit am Bildschirm. Alle diese Punkte sind durch änderungsfreundliche Entwicklungsplattformen auch in späten Projektphasen leicht behebbar.

Die Beiträge des Prototyping zur Sicherung der Akzeptanz durch die Beteiligung der Anwender wurden schon erörtert.

Die Beiträge von Prototyping und RAD zur Erhöhung der Anwenderzufriedenheit und zur Kostensenkung können unserer Meinung nach nicht hoch genug eingestuft werden. Wir sind schon im Begriff, uns daran zu gewöhnen und es selbstverständlich zu finden; dabei gibt es immer noch große Bereiche der EDV-Anwendung, in denen diese Möglichkeiten auch heute noch nicht zur Verfügung stehen oder nicht genutzt werden.

4. Was trotz Prototyping und RAD schwierig bleibt?

Prototyping und RAD erleichtern es, EDV-Systeme im Verlaufe ihrer Entwicklung zu ändern. Watzlawick u.a.¹ unterscheiden zwei Formen der Veränderung: "die eine findet innerhalb eines bestimmten Systems statt, das selbst unverändert bleibt, während das Eintreten der anderen das System selbst verändert" (a.a.O., S. 29). Sie sprechen von einem Wandel erster Ordnung bzw. einem Wandel zweiter Ordnung.

Betrachten wir eine EDV-Anwendung als ein System von Funktionen und Daten, die zu Applikationen zusammengefasst und auf bestimmten Hardware- und Betriebssystemplattformen implementiert werden (siehe die Grafik in Abschnitt 2.2), so erleichtern Prototyping und RAD den Wandel erster Ordnung. Ist es jedoch erforderlich, den Charakter des Systems insgesamt zu ändern, z.B. aus einem Batch-orientierten ein Dialog-orientiertes System oder aus einem Auskunftssystem ein Steuerungssystem zu machen, erleichtern uns dies Prototyping und RAD nur noch marginal. Gleiches gilt, wenn die für den Systembetrieb erforderlichen Daten nicht rechtzeitig oder nicht in genügender Genauigkeit zur Verfügung stehen.

Prototyping und RAD ersparen uns also nicht die Ausarbeitung einer durchdachten EDV-Architektur, sie erleichtern jedoch entscheidend den Wandel erster Ordnung, nämlich die Systemoptimierung im Rahmen einer gegebenen Architektur.

5. Gute Architektur kommt nicht von Gelegenheitsarchitekten

Softwareentwicklung ohne Programmierer? Softwareentwicklung durch die Anwender? Mag sein, es kommt aber auch darauf an, wen man als Programmierer und wen man als Anwender bezeichnet. Und es kommt sicher auch darauf an, wie komplex ein Softwaresystem ist und wieviel davon schon vorgegeben. Wenn wir uns darauf einigen, dass jemand der z.B. in Excel Makros schreibt oder vergleichbare Leistungen in anderen System erbringt, kein Programmierer ist, kann man den oben zitierten Verheißungen vielfach zustimmen. Wenn wir eine Analogie zum Bauen herstellen, so sehen wir, dass im Zeitalter der Fertighäuser sehr viele Bauherren keinen Architekten mehr brauchen. Allerdings sind die Fertighäuser von Architekten entworfen worden und die mehr oder minder hohe Qualität des Entwurfes erkennt jeder, der ein Fertighaus seinen individuellen Bedürfnissen anpassen will. Wenn wir noch einen Schritt weiter zur Städteplanung gehen, so sehen wir aber die Grenzen der Eigenplanung deutlich an den städtebaulichen Fleckerlteppichen mancher Gemeinden. Wenn wir also einer "Verhüttelung" der EDV-Landschaft entgehen wollen, so kommen wir nicht umhin, den Anwender auch im Zeitalter von Prototyping und RAD durch Profis zu betreuen und manchmal auch zu führen.

Die Pflichten des Anwenders enden also dort, wo es um die Gesamtarchitektur und die optimale technische Gestaltung des EDV-Systems geht, sie beschränken sich auf die Gestaltung des Umfeldes

des EDV-Einsatzes, wie wir es oben beschrieben haben. Aber dort wird der Erfolg des EDV-Einsatzes zum Großteil entschieden, egal ob wir mit Prototyping und RAD arbeiten oder nicht.